

BACKOUT TRANSACTION

BACKOUT [TRANSACTION]

Function

This statement is used to back out all database updates performed during the current logical transaction. This statement also releases all records held during the transaction.

The BACKOUT TRANSACTION statement is executed only if a database transaction under control of Natural has taken place. For which databases the statement is executed depends on the setting of the profile parameter ET (see your Natural Installation and Operations documentation):

- If ET=OFF, the statement is executed only for the database affected by the transaction.
- If ET=ON, the statement is executed for all databases that have been referenced since the last execution of a BACKOUT TRANSACTION or END TRANSACTION statement.

Note for ENTIRE SYSTEM SERVER:

This statement is not available with ENTIRE SYSTEM SERVER.

Considerations for DL/I Databases

Because PSB scheduling is terminated by a syncpoint request, Natural saves the PSB position before executing the BACKOUT TRANSACTION statement. Before the next command execution, Natural re-schedules the PSB and tries to set the PCB position as it was before the backout. The PCB position might be shifted forward if any pointed segment had been deleted in the time period between the backout and the following command.

Considerations for SQL Databases

As most SQL databases close all cursors when a logical unit of work ends, a BACKOUT TRANSACTION statement must not be placed within a database modification loop; instead, it has to be placed after such a loop.

Backout Transaction Issued by Natural

If the user interrupts the current Natural operation with a terminal command (command "%" or CLEAR key), Natural issues a BACKOUT TRANSACTION statement (see also the terminal command "%" in the Natural Reference documentation).

Additional Information

For additional information on the use of the transaction backout feature, see the section Database Access of the Natural Programming Guide.

Example

```

/* EXAMPLE 'BOTEX1S': BACKOUT TRANSACTION (STRUCTURED MODE)
/*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
2 NAME
2 DEPT
2 LEAVE-DUE
2 LEAVE-TAKEN
1 #DEPT (A6)
1 #RESP (A3)
END-DEFINE
/*****
LIMIT 3
INPUT 'DEPARTMENT TO BE UPDATED:' #DEPT
IF #DEPT = ' '
STOP
END-IF
/*****
FIND EMPLOY-VIEW WITH DEPT = #DEPT
IF NO RECORDS FOUND
REINPUT 'NO RECORDS FOUND'
END-NOREC
INPUT 'NAME:          ' NAME (AD=O) /
      'LEAVE DUE:      ' LEAVE-DUE (AD=M) /
      'LEAVE TAKEN:' LEAVE-TAKEN (AD=M)

UPDATE
END-FIND
/*****
INPUT 'UPDATE TO BE PERFORMED YES/NO:' #RESP
  DECIDE ON FIRST #RESP
    VALUE 'YES'
      END TRANSACTION
    VALUE 'NO'
      BACKOUT TRANSACTION
    NONE
      REINPUT 'PLEASE ENTER YES OR NO'
  END-DECIDE
/*****
END

```

DEPARTMENT TO BE UPDATED: **MGMT30**

NAME :	POREE
LEAVE DUE :	45
LEAVE TAKEN :	31

UPDATE TO BE PERFORMED YES/NO: NO
--

Equivalent reporting-mode example: See program BOTEX1R in library SYSEXRM.